What Is Claimed Is:

1. A method for detecting and preventing electronic fraud in electronic transactions between a client and a user, the method comprising:

generating a fraud detection and prevention model software component for using a plurality of intelligent technologies to determine whether information sent by the user to the client associated with a new electronic transaction is fraudulent, wherein the model software component is trained on a database of past electronic transactions provided by the client;

querying the model software component with a current electronic transaction to determine whether information sent by the user to the client associated with the current electronic transaction is fraudulent; and

updating the model software component with the current electronic transaction.

2. The method of claim 1, wherein the electronic transactions comprise web-based transactions and transactions conducted over wireless networks with the use of cellular phones.

3. The method of claim 1, wherein the plurality of intelligent technologies comprise one or more of the following: neural networks; multi-agents; data mining; case-based reasoning; rule-based reasoning;

fuzzy logic; constraint programming; and genetic algorithms.

4. The method of claim 1, wherein the fraud detection and prevention model software component comprises a plurality of sub-models, each sub-model implementing an intelligent technology to determine whether the electronic transaction is fraudulent.

5. The method of claim 4, wherein the fraud detection and prevention model software component comprises a binary file for implementing the plurality of sub-model software components.

6. The method of claim 1, wherein training the model software component on a database of past electronic transactions provided by the client comprises training the plurality of sub-model software components and creating the binary file for implementing the plurality of sub-model software components.

7. The method of claim 1, wherein the database of past electronic transactions comprises a plurality of tables, wherein each table comprises a plurality of data fields and data records associated with a plurality of electronic transactions.

8. The method of claim 1, wherein querying the model software component with a current electronic transaction to determine whether information sent by the user to the client associated with the current electronic transaction is fraudulent comprises providing the information as input to the binary file and running the binary file to generate a binary output decision on whether the electronic transaction is fraudulent or not.

9. The method of claim 8, wherein running the binary file to generate an output decision on whether the electronic transaction is fraudulent comprises running the plurality of sub-model software components to generate a plurality of sub-model decisions and combining the plurality of sub-model decisions to generate the output decision.

10. The method of claim 9, wherein combining the plurality of sub-model decisions to generate the output decision comprises assigning a vote to each sub-model decision and generating the output decision based on the majority of votes determining whether the electronic transaction is fraudulent or not.

11. The method of claim 9, wherein combining the plurality of sub-model decisions to generate the output decision comprises assigning a weighted vote to each one of the sub-models, wherein the weighted vote is assigned to prioritize the sub-model decisions, and generating the output decision based on the highest

number of votes determining whether the electronic transaction is fraudulent or not.

12. The method of claim 9, wherein combining the plurality of sub-model decisions to generate the output decision comprises providing a plurality of meta-rules to determine how the sub-model decisions are combined to generate the output decision.

13. The method of claim 1, wherein updating the model software component with the current electronic transaction comprises updating the binary file without retraining the model.

14. The method of claim 1, wherein updating the model software component with the current electronic transaction further comprises updating the database with the current electronic transaction and retraining the model to generate a new binary file.

15. A system for dynamic detection and prevention of fraud in electronic transactions between a client and a user, the system comprising:

a fraud detection and prevention model software component for determining the probability that information sent by the user to the client associated with a current electronic transaction is fraudulent, wherein the model software component comprises a plurality of integrated sub-model components, each sub-

model component comprising software routines for implementing an intelligent technology;

a fraud detection and prevention model training software component for training the model on a database of past electronic transactions provided by the client and for updating the model with data from the current electronic transaction; and

a fraud detection and prevention model querying component for querying the fraud detection and prevention model with the current electronic transaction to determine the likelihood that the current electronic transaction is fraudulent.

16. The system of claim 15, wherein the electronic transactions comprise web-based transactions and transactions conducted over wireless network with the use of cellular phones.

17. The system of claim 15, wherein the plurality of sub-model software components comprises one ore more of the following: a neural network software component; a data mining software component; a multi-agent software component; a case-based reasoning software component; a rule-based reasoning software component; a constraint programming software component; and a genetic algorithms software component.

18. The system of claim 17, wherein the neural network software component comprises a neural network software routine for generating a neural network for predicting an output from an input data record

associated with the current electronic transaction, the output determining the likelihood that the current electronic transaction is fraudulent.

19. The system of claim 18, wherein the neural network comprises a plurality of interconnected processing elements arranged into three layers, the layers comprising an input layer; a hidden layer; and an output layer.

20. The system of claim 19, wherein each processing element in the input layer represents a field of the input data record.

21. The system of claim 19, wherein each processing element in the output layer represents a predicted value for the output of the input data record.

22. The system of claim 19, wherein each processing element in the hidden layer comprises:

a plurality of inputs, each input connected to each processing element in the input layer;

a plurality of input weights, each input weight associated with an input from the plurality of inputs;

a first and a second limiting threshold;

a plurality of output weights, each output weight associated with an output from the plurality of outputs; and

a plurality of outputs, each output connected to each processing element in the output layer.

23. The system of claim 18, wherein predicting an output from an input data record associated with the current electronic transaction comprises propagating the input data record through the plurality of interconnected processing elements in the neural network.

24. The system of claim 23, wherein propagating the input data record through the plurality of interconnected processing elements in the neural network comprises the steps of:

a) determining the distance between the input data record and the plurality of input weights for a single processing element in the hidden layer;

b) verifying if the distance between the input data record and the plurality of input weights is between the first and second limiting thresholds for the single processing element in the hidden layer;

c) adding the plurality of input weights to the plurality of output weights of each processing element in the hidden layer if the distance is between the first and second limiting thresholds of the single processing element in the hidden layer;

d) repeating steps a), b), and c) for each processing element in the hidden layer; and

f) determining the processing element in the output layer connected to the output of the processing element in the hidden layer from the plurality of

processing elements in the hidden layer that has the higher output weight; and

g) assigning the processing element in the output layer connected to the output of the processing element in the hidden layer that has the highest output weight to the predicted output value of the input record.

25. The system of claim 24, wherein the distance between the input data record and the plurality of input weights for a single processing element in the hidden layer is computed with a distance measure selected from a plurality of distance measures, comprising:

an Euclidean distance measure;

a Manhattan distance measure;

a Normalized Euclidean distance measure;

a Normalized Manhattan distance measure; and

a Weighted-Euclidean distance measure.

26. The system of claim 17, wherein the data mining software component comprises software routines for creating a decision tree to group a plurality of data records in the database together according to their similarity.

27. The system of claim 26, wherein creating the decision tree comprises the steps of:

a) forming a subset containing all the data records in the database;

b) splitting the subset into a first and second subset according to an impurity function;

c) assigning the first and the second subsets to a first and a second node in the binary decision tree;

d) determining whether the data records in the first and the second subsets belong to the same output class;

e) repeating steps b), c), and d) for each of the first and second subsets if the data records in each of the first and second subsets do not belong to the same output class.

28. The system of claim 27, wherein the impurity function comprises an entropy function and a Gini index function to determine the information value of each field in the database.

29. The system of claim 27, wherein splitting the subset into a first and a second subset according to an impurity function comprises:

determining the field with the highest information value;

determining the data records in the subset that satisfy a test on the field with the highest information value;

assigning the data records in the subset that satisfy the test to the first subset; and

assigning the data records in the subset that do not satisfy the test to the second subset.

30. The system of claim 29, wherein the test on the field with the highest information value comprises a test on whether the value of the field is smaller than a given value if the field is a numeric field, or equal to a given value if the field is a symbolic field.

31. The system of claim 30, wherein the given value is selected to split the data records in the subset into a first and a second subsets containing the highest possible number of data records belonging to the same output class.

32. The system of claim 17, wherein the multi-agent software component comprises a software agent for creating intervals of normal values for fields in the database and a software agent for determining dependencies between each field in the database to determine the likelihood that the electronic transaction is fraudulent.

33. The system of claim 17, wherein the case-based reasoning software component comprises software routines for:

creating a plurality of sample cases to represent all the data records in the database; and

searching the decision tree to determine the similarity between a data record associated with a current electronic transaction and the sample cases; and

determining whether the electronic transaction is fraudulent based on the sample case that has the highest similarity to the data record associated with the current electronic transaction.

34. The system of claim 17, wherein the rule-based software component comprises using a plurality of business rules to determine whether the data record associated with the current electronic transaction is fraudulent.

35. The system of claim 17, wherein the constraint programming software component comprises using a plurality of constraints to determine whether the data record associated with the current electronic transaction is fraudulent.

36. The system of claim 17, wherein the fuzzy logic software component comprises fuzzifying a plurality of rules to generate a plurality of fuzzy rules to determine whether the data record associated with the current electronic transaction is fraudulent.

37. The system of claim 15, wherein the model training software component comprises a model training software interface and a plurality of software routines for training each one of the sub-models.

38. The system of claim 37, wherein the model training interface comprises a graphical user interface comprising a plurality of dialog boxes and forms for the client to perform a plurality of steps to configure, create, and test the model software component, the steps comprising:

configuring the model parameters;

visualizing the contents of a database containing data records of past electronic transactions used for training the model software component;

retrieving statistics of the data in the database;

creating the model software component and saving it into a binary file; and

creating a web query form for querying the model software component on the web with data associated with an electronic transaction.

39. The system of claim 38, wherein the steps further comprise testing the model software component.

40. The system of claim 38, wherein configuring the model parameters comprises:

selecting the database containing a plurality of tables of data records of past electronic transactions;

selecting a plurality of tables from the plurality of tables in the database to be used for training the model software component;

selecting a field in the database to be designated as the model output;

selecting the values that are considered normal values for the model output;

selecting the sub-model software components to be used in the model.

41. The system of claim 37, wherein the plurality of software routines for training each one of the sub-models comprises one or more of the following: a neural network training routine; a data mining training routine; a multi-agent training routine; a case-based reasoning training routine; a rule-based reasoning training routine; a fuzzy logic training routine; a constraint programming training routine; and a genetic algorithm training routine.

42. The system of claim 41, wherein the neural network training routine comprises the steps of:

a) initializing a training set containing all the data records in the database;

b) creating a distance matrix D containing the distance between two data records in the training set;

c) sorting the distance matrix D;

d) determining a subset S of data records in each row of the distance matrix that contains the highest number of data records having the same output;

e) adding a new processing element to the hidden layer of the neural network having a first and a second limiting thresholds;

f) removing the subset S from the training set;

g) adjusting the weights in the neural network; and

h) repeating steps b), c), d), e), f), and g) until the training set is empty.

43. The system of claim 42, wherein the distance between two data records in the training set is computed with a distance measure selected from a plurality of distance measures, comprising:

an Euclidean distance measure;

a Manhattan distance measure;

a Normalized Euclidean distance measure;

a Normalized Manhattan distance measure; and

a Weighted-Euclidean distance measure.

44. The system of claim 41, wherein the first limiting threshold corresponds to the smallest distance in the subset S and the second limiting threshold corresponds to the highest distance in the subset S.

45. A system for creating a model for detecting and preventing fraud in electronic transactions, the system comprising:

a general purpose computer;

a memory, the memory storing a plurality of software routines for execution by the processor, wherein the plurality of software routines comprise:

a model training interface software routine;

a plurality of software routines for creating a plurality of user-selectable sub-models that use an intelligent technology to detect and predict electronic fraud; and

a software routine for combining the plurality of user-selectable sub-models to create the model for detecting and preventing electronic fraud.

46. The system of claim 45, wherein the electronic transactions comprise web-based transactions and transactions conducted over wireless networks with the use of cellular phones.

47. The system of claim 45, wherein the plurality of software routines comprises one or more of the following: a neural network training software routine; a multi-agent training software routine; a data mining training software routine; a case-based reasoning training software routine; a rule-based reasoning training software routine; a fuzzy logic software routine; a constraint programming software routine; and a genetic algorithm software routine.

48. The system of claim 45, wherein creating the plurality of user-selectable sub-models comprises training the sub-models on a database of past electronic transactions.

49. The system of claim 45, wherein the model training interface software routine comprises a graphical user interface comprising a plurality of dialog boxes and forms for the client to perform a plurality of steps to configure, create, and test the model, the steps comprising:

configuring the model parameters;

visualizing the contents of the database containing data records of past electronic transactions used for training the model;

retrieving statistics of the data in the database;

creating the model and saving it into a binary file; and

creating a web query form for querying the model on the web with data associated with an electronic transaction.

50. The system of claim 49, wherein the steps further comprise testing the model.

51. The system of claim 49, wherein configuring the model parameters comprises:

selecting the database containing a plurality of tables of data records of past electronic transactions;

selecting a plurality of tables from the plurality of tables in the database to be used for training the sub-models;

selecting a field in the database to be designated as the model output;

selecting the values that are considered normal values for the model output;

selecting the sub-models to be used in the model.

52. The system of claim 45, wherein combining the plurality of sub-models to create the model for detecting and preventing electronic fraud comprises generating a binary output decision to determine whether the current electronic transaction is fraudulent based on a plurality of sub-model decisions, wherein generating the binary output decision comprises assigning a vote to each sub-model decision and generating the output decision based on the majority of votes determining whether the electronic transaction is fraudulent or not.

53. The system of claim 45, wherein combining the plurality of sub-models to create the model for detecting and preventing electronic fraud comprises generating a binary output decision to determine whether the current electronic transaction is fraudulent based on a plurality of sub-model decisions, wherein generating the binary output decision further comprises assigning a weighted vote to each one of the sub-models, wherein the weighted vote is assigned to prioritize the sub-model decisions, and generating the output decision

based on the highest number of votes determining whether the electronic transaction is fraudulent or not.

54. The system of claim 45, wherein combining the plurality of sub-models to create the model for detecting and preventing electronic fraud comprises generating a binary output decision to determine whether the current electronic transaction is fraudulent based on a plurality of sub-model decisions, wherein generating the binary output decision further comprises providing a plurality of meta-rules to determine how the sub-model decisions are combined to generate the output decision.

55. A fraud detection and prevention model software component for detecting and preventing electronic fraud in electronic transactions, the model comprising:

a plurality of integrated sub-model software components, each sub-model software component comprising software routines for implementing an intelligent technologies to determine whether the electronic transaction is fraudulent; and

a voting and arbitrating software routine for combining the individual predictions of the plurality of sub-model components into a binary decision determining whether the electronic transaction is fraudulent.

56. The fraud detection and prevention model of claim 55, wherein the model comprises a binary file for implementing the plurality of sub-model software components.

57. The fraud detection and prevention model of claim 55, wherein the plurality of sub-model software components comprise one or more of the following: a neural network software component; a multi-agent software component; a data mining software component; a case-based reasoning software component; a rule-based reasoning software component; a fuzzy logic software component; a constraint programming software component; and a genetic algorithms software component.

58. The fraud detection and prevention model of claim 55, wherein the voting and arbitrating software routine comprises assigning a vote to the individual predictions of each sub-model and generating the binary decision based on the majority of votes determining whether the electronic transaction is fraudulent or not.

59. The fraud detection and prevention model of claim 55, wherein the voting and arbitrating software routine further comprises assigning a weighted vote to the individual predictions of each sub-model, wherein the weighted vote is assigned to prioritize the individual predictions, and generating the binary decision based on the highest number of votes determining whether the electronic transaction is fraudulent or not.

60. The fraud detection and prevention model of claim 55, wherein the voting and arbitrating software routine further comprises providing a plurality of meta-rules to determine how the individual predictions are combined to generate the output decision.

61. A method for dynamic detection and prevention of network intrusion, the method comprising:

providing a fraud detection and prevention model software component comprising a plurality of sub-model software components, each sub-model software component implementing an intelligent technology to determine whether data associated with a current network user is fraudulent, wherein the sub-model software components are trained on a database of past network usage profiles;

querying the model software component with data associated with a current network user to determine whether there is network intrusion; and

updating the model software component.

62. The method of claim 61, wherein the fraud detection and prevention model software component comprises detecting and preventing electronic fraud in electronic transactions.

63. The method of claim 62, wherein the electronic transactions comprise web-based transactions and transactions conducted over wireless networks with the use of cellular phones.

64. The method of claim 61, wherein the plurality of sub-model software components comprises one or more of the following: a neural network software component; a multi-agent software component; a data mining software component; a case-based reasoning software component; a rule-based reasoning software component; a fuzzy logic software component; a constraint programming software component; and a genetic algorithms software component.

65. The method of claim 61, wherein the fraud detection and prevention model software component comprises a binary file for implementing the plurality of sub-model software components.

66. The method of claim 61, wherein the database comprises a plurality of tables, wherein each table comprises a plurality of data fields and data records associated with a plurality of network usage profiles.

67. The method of claim 61, wherein querying the model software component with data associated with a current network user to determine whether there is network intrusion comprises providing data associated a current network user as input to the binary file and running the binary file to generate a binary output decision on whether there is network intrusion or not.

68. The method of claim 67, wherein running the binary file to generate an output decision on whether there is network intrusion comprises running the plurality of sub-model software components to generate a plurality of sub-model decisions and combining the plurality of sub-model decisions to generate the output decision.

69. The method of claim 68, wherein combining the plurality of sub-model decisions to generate the output decision comprises assigning a vote to each sub-model decision and generating the output decision based on the majority of votes determining whether there is network intrusion or not.

70. The method of claim 68, wherein combining the plurality of sub-model decisions to generate the output decision further comprises assigning a weighted vote to each one of the sub-model software components, wherein the weighted vote is assigned to prioritize the sub-model decisions, and generating the output decision based on the highest number of votes determining whether there is network intrusion or not.

71. The method of claim 68, wherein combining the plurality of sub-model decisions to generate the output decision further comprises providing a plurality of meta-rules to determine how the sub-model decisions are combined to generate the output decision.

72. The method of claim 61, wherein updating the model software component with data associated with a current network user comprises updating the binary file without retraining the model software component.

73. The method of claim 61, wherein updating the model software component with data associated with a current network user further comprises updating the database and retraining the model software component to generate a new binary file.

74. A system for dynamic detection and prevention of network intrusion, the system comprising:

a network intrusion detection and prevention model software component, wherein the model software component comprises a plurality of integrated sub-model components, each sub-model component comprising software routines for implementing an intelligent technology; and

a network intrusion model training software component for training the model on a database containing data corresponding to past network intrusions and for updating the model with data from potential network intrusions; and

a network intrusion model querying component for querying the network intrusion model to determine whether a user is illegally breaching the security of the network.

75. The system of claim 74, wherein the plurality of sub-model software components comprises one ore more of the following: a neural network software component; a data mining software component; a multi-agent software component; a case-based reasoning software component; a rule-based reasoning software component; a constraint programming software component; and a genetic algorithms software component.

76. The system of claim 74, wherein the model training software component comprises a model training software interface and a plurality of software routines for training each one of the sub-models.

77. The system of claim 76, wherein the model training interface comprises a graphical user interface comprising a plurality of dialog boxes and forms for the client to perform a plurality of steps to configure, create, and test the model software component, the steps comprising:

configuring the model parameters;

visualizing the contents of a database containing data records of past network usage profiles used for training the model;

retrieving statistics of the data in the database;

creating the model software component and saving it into a binary file; and

creating a web query form for querying the model software component on the web with data associated with past network usage profiles.

78. The system of claim 77, wherein the steps further comprise testing the model software component.

79. The system of claim 77, wherein configuring the model parameters comprises:

selecting the database containing a plurality of tables of data records of past network usage profiles;

selecting a plurality of tables from the plurality of tables in the database to be used for training the model software component;

selecting a field in the database to be designated as the model output;

selecting the values that are considered normal values for the model output;

selecting the sub-model software components to be used in the model software component.

80. The system of claim 77, wherein the plurality of software routines for training each one of the sub-model software components comprises one or more of the following: a neural network training routine; a data mining training routine; a multi-agent training routine; a case-based reasoning training routine; a rule-based reasoning training routine; a fuzzy logic

training routine; a constraint programming training routine; and a genetic algorithm training routine.

81. A system for creating a model for detecting and preventing network intrusion, the system comprising:

a general purpose computer connected to the network;

a memory, the memory storing a plurality of software routines for execution by the processor, wherein the plurality of software routines comprise:

a model training interface software routine;

a plurality of software routines for creating a plurality of sub-models that use an intelligent technology to detect and prevent network intrusion; and

a software routine for combining the plurality of sub-models to create the model for detecting and preventing network intrusion.

82. The system of claim 81, wherein the plurality of software routines comprises one or more of the following: a neural network training software routine; a multi-agent training software routine; a data mining training software routine; a case-based reasoning training software routine; a rule-based reasoning training software routine; a fuzzy logic software routine; a constraint programming software routine; and a genetic algorithm software routine.

83. The system of claim 81, wherein creating the plurality of sub-models comprises training the sub-models on a database of network usage profiles.

84. The system of claim 81, wherein the model training interface software routine comprises a graphical user interface comprising a plurality of dialog boxes and forms for the client to perform a plurality of steps to configure, create, and test the model, the steps comprising:

configuring the model parameters;

visualizing the contents of the database containing data records of past network usage profiles used for training the model;

retrieving statistics of the data in the database;

creating the model and saving it into a binary file; and

creating a web query form for querying the model on the web with data associated with a current network user.

85. The system of claim 84, wherein the steps further comprise testing the model.

86. The system of claim 84, wherein configuring the model parameters comprises:

selecting the database containing a plurality of tables of data records of past network usage profiles;

selecting a plurality of tables from the plurality of tables in the database to be used for training the sub-models;

selecting a field in the database to be designated as the model output;

selecting the values that are considered normal values for the model output;

selecting the sub-models to be used in the model.

87. The system of claim 81, wherein combining the plurality of sub-models to create the model for detecting and preventing network intrusion comprises generating a binary output decision to determine whether there is network intrusion based on a plurality of sub-model decisions, wherein generating the binary output decision comprises assigning a vote to each sub-model decision and generating the output decision based on the majority of votes determining whether there is network intrusion or not.

88. The system of claim 81, wherein combining the plurality of sub-models to create the model for detecting and preventing network intrusion comprises generating a binary output decision to determine whether there is network intrusion based on a plurality of sub-model decisions, wherein generating the binary output decision further comprises assigning a weighted vote to

each one of the sub-models, wherein the weighted vote is assigned to prioritize the sub-model decisions, and generating the output decision based on the highest number of votes determining whether there is network intrusion or not.

89. The system of claim 81, wherein combining the plurality of sub-models to create the model for detecting and preventing network intrusion comprises generating a binary output decision to determine whether there is network intrusion based on a plurality of sub-model decisions, wherein generating the binary output decision further comprises providing a plurality of meta-rules to determine how the sub-model decisions are combined to generate the output decision.